A METHOD OF MAKING SECURE THE TRANSMISSION OF A MESSAGE FROM AN EMITTER DEVICE TO A RECEIVER DEVICE

FIELD OF THE INVENTION

The invention relates to a secure method of transmitting messages from an emitter device to a receiver device.

BACKGROUND OF THE INVENTION

When information is transmitted from an emitter device to a receiver device, the information as contained in a message is in danger of being degraded in transit. Such degradation can stem from a defect in the emitter, in the transmission path, or in the receiver of the message, or it can stem from third party fraud. The message as received is then corrupted.

That is why methods have been developed that make it possible to verify that transmitted messages are not corrupted.

Furthermore, when information is transmitted from an emitter device to a receiver device, it is sometimes useful to make the message confidential so that access to said information is reserved for a limited number of people only, in general to the emitter and to the receiver of the message.

That is why methods have been developed enabling message confidentiality to be preserved.

Finally, when information contained in a message is transmitted to a receiver device, it is often useful to be able to authenticate the message as coming genuinely from the emitter device.

That is why message-authentication methods have been developed.

Known methods for verifying lack of corruption, for preserving confidentiality, and for providing authentication, i.e. methods for making messages secure, generally consist in encrypting the message and in joining a certificate thereto prior to transmission. The receiver device then decrypts the message, verifies the certificate, and possibly, if the message is a computer program, executes it.

Such methods are clearly somewhat cumbersome insofar as decrypting and verifying the certificate comprise two distinct operations. This is particularly true when the operations of encrypting and of decrypting are lengthy.

OBJECTS AND SUMMARY OF THE INVENTION

a1> ~~In the light of the above, a problem that the invention seeks to resolve is that of providing a method~~ of making secure the transmission of a message from an emitter device to a receiver device in which it is not necessary to implement the above-mentioned two steps of decrypting the message and of verifying the certificate.

a2> ~~In the light of the problem as posed above, the~~ invention provides a method of making secure the transmission of a message from an emitter device to a receiver device, the method being characterized in that:

- the message is subdivided into n elementary units, where n is a number greater than 1;
- a logical property is defined in such a manner that for any elementary unit, the logical property when applied to an authentic elementary unit gives a logical value of the type true;
- the message is encrypted by encryption means of the emitter device using an encryption algorithm having a key so as to obtain an encrypted result;
- the encrypted result is transmitted by the emitter device to the receiver device;
- the encrypted result is decrypted by the receiver device using a decryption algorithm having a secret key so as to obtain a decrypted result;
- the decrypted result is subdivided into elementary units;
- the logical property is applied to the elementary units so as to obtain, for each unit, a logical value of the type true or of the type false; and
- the message is considered as being authentic and uncorrupted providing the logical value of each unit is of the type true.

~~Advantageously, the message is then stored.~~

It will also be observed that, advantageously, the message Prgm is a computer program suitable for being executed and/or interpreted by the receiver device R. The elementary units are instructions of the program Prgm. The property P as applied to an elementary unit I gives a logical value of the type true whenever the elementary unit I is executable and/or interpretable. The property P as applied to an elementary unit I gives a logical value of the type false whenever the elementary unit I is not executable and/or interpretable. The receiver device R is a portable object having a memory, of the smart card type. The receiver device R includes a portable object having a memory, of the smart card type. The portable object having a memory is a subscriber identity module (SIM). The message Prgm is written in a high level interpreted language. The high level language is the Java language. The computer program is made up of a set of precompiled instructions. The message Prgm is encrypted as a continuous flow or in chained-together blocks. The message Prgm is encrypted in blocks, and the blocks of the encrypted message Prgm are permutated. One of the permutated blocks is a starting block or an end block of the message Prgm. The result Kc(Prgm) is decrypted in blocks, each encrypted block giving rise to a decrypted block which occupies the same space as the encrypted block. The encryption and decryption algorithms make use of a random number transmitted by the emitter device E to the receiver device R. The message Prgm is recorded, after verification, in a non-volatile memory of the receiver device R.

The invention will be better understood on reading the following non-limiting description.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the invention, the message Prgm is transmitted from an emitter device E to a receiver device R.

By way of example, the message Prgm is a computer program suitable for being executed and/or interpreted.

By way of example, the emitter device E is a server, a computer, a transmitter station in a telecommunications network, or a contact or contactless smart card reader, i.e. any device capable of encrypting and emitting a message. Naturally, the emitter device E must be considered in the broad sense as including composite devices made up of physical portions that are in fact separate, one portion serving for example to encrypt the message and another actually to emit said message.

By way of example, the receiver device R is a computer optionally provided with a smart card reader and a card inserted in said reader, a receiver station in a telecommunications network, a portable telephone optionally fitted with a subscriber identity module (SIM), or even a smart card or such a module, in other words any device capable of receiving a message or of storing the message, and advantageously in the event of the message being a computer program, capable of interpreting and/or executing the program. When the receiver device advantageously comprises a portable memory object of the smart card type, the portable object can be a payment card or a card controlling access, e.g. to a computer network.

The following description of the invention is limited to examples in which the message is a computer program Prgm.

In the invention, the computer program Prgm is subdivided into n elementary units I, where n is an integer greater than or equal to 1. It comprises instructions, blocks of instructions, or in the case of the program Prgm being written in interpretable languages of the Java type, precompiled program instructions (or bytecodes).

In the invention, a logical property P is defined in such a manner that for every elementary unit I, this property P when applied to an authentic elementary unit gives a logical value P(I) of the true type.

Nevertheless, attempts are made to find a property P which, when applied to an elementary unit I, gives a logical value P(I) of the false type whenever said elementary unit I has been modified and corresponds, for example, to an unrecognizable program instruction, in particular one that is not capable of being interpreted and/or executed.

In the invention, the program Prgm is encrypted by encryption means of the emitter device E using an encryption algorithm having a key Kc known to said device E so as to obtain a result Kc(Prgm). Encryption guarantees that the program Prgm is kept confidential during emission and reception, and above all while it is in transit to the receiver device R. The result Kc(Prgm) is thus transmitted from the device E to the receiver device R.

It is then decrypted by the device R using an encryption algorithm including a secret key Kd that is known to the receiver device. The decrypted result Kd(Kc(Prgm)) is then obtained.

This key Kc can be specific to the device E and known to the device R, or specific to the device R and also known to the device E. An example of the first configuration is the case where the device R is a subscriber to a service delivered by the emitter device. An example of the second configuration is the case where the receiver device, on requesting the transmission of a program, delivers the key Kc, while the decryption key Kd remains known to the receiver device only. Another example of the same configuration is the case where Kc and Kd are identical (private key system) and where said key is sent in encrypted form by the receiver device to the emitter device.

In the invention, the decrypted result Kd(Kc(Prgm)) is subdivided or decomposed into n elementary units that are images of, or that correspond to the n elementary

units that resulted from subdividing the program Prgm in the emitter device E.

The logical property P is then applied to said n elementary units so as to obtain for each unit a logical value of the type true or of the type false.

When all of the logical values are of the type true, it is highly probable that the decrypted program is identical to the encrypted program and that the key that was used for encryption was the expected key Kc. The receiver device R then deduces that the program Prgm is not corrupt and that it was indeed emitted by an emitter device E possessing the key Kc, and is therefore authentic.

However, when at least one logical value is of the type false, the decrypted program is different from Prgm and the receiver device R deduces that the program Prgm has been subject to at least one modification on emission, on reception, or in transit, and/or that said program Prgm was encrypted into the message using a key other than Kc, i.e. an unexpected key. The program is then neither uncorrupted nor authentic.

The invention thus makes it possible in a single operation of encryption and decryption to guarantee simultaneously that the message is uncorrupted, that it is authentic, and that the program Prgm has been kept confidential.

It is assumed, by way of example, that the instructions of the computer language in which the program Prgm is written are instructions that are encoded on four bytes, giving a theoretical total of 2^{32} possible codes for defining an instruction. Naturally, some codes defined by a set of parameters do not correspond to any comprehensible instruction. In addition, certain parameters for certain codes, typically the last three bytes, have only certain values that are authorized. Thus, a memory address cannot be negative or cannot lie outside the space allocated to the program Prgm. That is

why the property P advantageously includes a parameter test, said test depending on instruction type.

If a unit non-detection rate C is defined as being the percentage of possible instructions that are not recognized as being false by applying the property P on decryption and following a single change to the program Prgm, the probability prob that the receiver device R will fail to detect the fraud, assuming that the single change is the cause of a change in each instruction of the decrypted result is given by:

$$prob = (1 - C)^n$$

For the following typical values, the following probabilities prob are obtained:

n	C (%)	prob
256	10%	1.9E-12
128	10%	1.4E-06
512	5%	3.9E-12
128	5%	1.4E-03

It is observed that the probability of a change, and in particular of a fraudulent change, passing through unobserved is very low, except for programs that have few instructions and for which the unit non-detection rate C is very high. This probability is <u>a fortiori</u> very low in the event of the program being encrypted using a key other than Kc.

Compared with conventional encryption operations, applying the property P does not require excessive investment, particularly in terms of excessive computation time. It enables errors to be detected in all types of program Prgm providing the encryption algorithm is of good quality, given the pseudo-random nature of any attempt at decrypting a falsified run of instructions.

The encrypting algorithm is advantageously of the chained block or continuous flow type. Thus, modifying any one elementary instruction will give rise to modifications in other instructions. In contrast, if the algorithm operates solely in blocks, the encrypted program can be decomposed as a run of n blocks, for example, necessarily corresponding to the n elementary units. By modifying one block and observing the behavior of the receiver device, the probability Prgm that the modification passes undetected is then equal to 1 - C, and is thus very high.

In order to avoid directed modification applied to the head block or the tail block of the encrypted program, the blocks of the encrypted program are permutated, for example, so that said head and tail blocks are at locations that cannot be predicted by a dishonest person, even though they are known to the devices E and R.

Confidentiality is also improved when the encryption algorithm makes use of a random number generated, e.g. by the receiver device R and communicated to the emitter device E. By way of example, this can be based on an EXCLUSIVE-OR operation applied to a determined number of bytes of the program or to all of it prior to encryption.

Finally, at the beginning and/or the end of the program, it is possible to insert, prior to encryption, empty instructions (NOPs) which the receiver device will recognize by applying the property P, and will then skip.

In a first implementation of the invention, the emitter device E is a base station of a GSM telecommunications network (where GSM = Global System for Mobile communications) or any other type of mobile telephone system that uses a security module, the receiver device R being a subscriber identity module SIM associated with a mobile telephone. The program Prgm for downloading into said SIM is encoded in the form of

precompiled instructions (bytecodes), e.g. written in the Java language.

Naturally, the invention applies in the same manner to other smart card systems, for example payment systems or access control systems.

In this first embodiment of the invention, the program is divided into n elementary units, one elementary unit being a precompiled instruction having a determined number of bits (either fixed or depending on the type of the instruction).

The logical property P is defined in such a manner that it takes a true logical value when the elementary unit to which it is applied is an executable instruction (or an interpretable instruction) or corresponds to a NOP instruction.

The program Prgm is then encrypted by the emitter device E using an encryption algorithm, e.g. of the RSA type (RSA = Rivet, Shamir, and Adelman) as described in US patent 4 405 829. An encryption result Kc(Prgm), i.e. a function of the key Kc, is then obtained.

This result Kc(Prgm), i.e. the encrypted program, is delivered by the base station to a transmitter station associated therewith and on to mobile telephone receiver means. It is then loaded into the card where it is recorded in non-volatile memory (EEPROM) prior to applying the decryption operation, given the length of time required to implement this operation in a SIM.

The result Kc(Prgm) is then decrypted using a decryption algorithm that includes a secret key Kd. Each block of the decrypted result is stored in the EEPROM of the SIM at the address of the corresponding encrypted result block. As a result, the memory space used for implementing decryption in accordance with the invention is minimized. It will be observed that in a variant implementation of the invention, with the help of at least one block's worth of available memory space, it is possible to store the decrypted result blocks at memory

addresses that are different from those of the encrypted blocks to which they correspond. Circular permutation is also possible, thereby improving security for the program during the decrypting step.

The property P is preferably applied after the encrypted result Kc(Prgm) has been fully decrypted, with the final result (program accepted or refused) being obtained only after all of the verifications have been performed. Thus, a dishonest person cannot merely detect which elementary unit I is recognized as giving rise to a false logical value when applying the property P.

Given the small amount of memory available in the SIM, a simple computation function is implemented for the property P. This is a function implemented by the interpreter itself. Once the encrypted result has been decrypted, the interpreter interprets the decrypted result by looking to see whether the instructions are meaningful or not. In other words, the interpreter analyzes the program in the same manner as it would while interpreting it in normal manner, with the exception that this interpretation is not followed by any effect other than verifying whether the decrypted result does indeed correspond to a program Prgm.

In a second implementation of the invention, the emitter device E is a server including a precompiled and encrypted form Kc(Prgm) of a program Prgm, e.g. written in the Java language. The receiver device R is a personal computer which is advantageously provided with a smart card reader into which a card is inserted. The personal computer has a hard disk and a memory zone that is secure, i.e. that cannot be read or written by third parties, for use in temporary or permanent storage of the decrypted results Kd(Kc(Prgm)) and of the keys. The computer also has software for loading programs Prgm and referred to as a "loader", which program is called each time it is necessary to load a precompiled program Prgm prior to said program Prgm being used (interpreted or

executed). In this second implementation of the invention, this software includes a decryption function which advantageously has the functional elements necessary for decryption and in particular the elements of the decryption algorithm. The loader software for loading programs is then said to be "overloaded". Naturally, other functional elements required for decryption can be contained in a non-volatile memory of the smart card. These elements are then called by the program loader software and the decryption function. Thus, the loader software makes it possible when associated with the card to decrypt the result Kc(Prgm) and to verify the decrypted result Kd(Kc(Prgm)) prior to interpreting said decrypted result Kd(Kc(Prgm)), i.e. once the property P has been applied with success to the program Prgm, after which the program Prgm can be executed.

The time and memory space constraints mentioned when describing the first implementation of the method of the invention are less important in this second implementation given that the card is used in this case solely as a secure physical medium for containing one or more keys or elements, e.g. tables, required for decryption. The card can even contain the entire secret decryption algorithm.

As a result, the property P need not merely be of the type described above, or else be a special property implemented in the verification algorithm. In one example, the verification algorithm verifies the precompiled instructions each time that a block of instructions from the encrypted result has been decrypted.

The data interchange stages between the card and the personal computer provided with the interpreter, the loader device, and associated with a card reader in which the card is inserted can then comprise three stages: an

initialization stage; a transfer stage; and a decryption/verification stage.

The initialization stage is a stage during which both a public key and a secret key are exchanged. This stage is launched during initialization of the decryption process. The pairs of keys are not written on the hard disk of the personal computer and can be recalculated at any time. During this stage, a reinitialization instruction is transmitted by the personal computer to the card. The computer then calculates a pair comprising a public key PKc and a secret key PKd, and then calculates a signature from the public key PKc using the secret key PKd. This signature is transmitted with the public key PKc to the card. It is then verified by the card using the public key PKc. The card then uses a secret key CKd to calculate a signature of the public key CKc. This signature is transmitted using the public key CKc to the personal computer. The computer verifies the signature using the public key CKc.

The transfer stage is a stage during which secret information is loaded from the card into the personal computer. This information enables the computer to decrypt the precompiled and encrypted form of the program Prgm. During this stage, the computer asks the card to transfer the secret decryption key Kd that it contains in its memory. The card encrypts this key using the key PKc and sends it to the computer. The computer decrypts this message using its key Kd, thus giving it the key Kc. It is then possible for the computer to decrypt the program Kc(Prgm) to obtain a program Prgm', which will be identical to the original program Prgm providing no attempt at fraud has occurred.

As this moment, the computer can subdivide the program Prgm' into elementary units, and can apply the property P thereto, as in the first implementation. If the result is satisfactory, said program is archived, e.g. on the hard disk. The computer can also calculate

verification information (e.g. a checksum, or better a hash total) and can archive it in the card memory in order to make it possible subsequently to verify that the program is not corrupted.